CL21: More OOP Practice + a 🪄magic🪄 method

Memory diagram

```python
class Pizza:
    size: str
    extra_cheese: bool
    toppings: int

    def __init__(self):
        """Initialize new Pizza instance"""
        self.size = "medium"
        self.extra_cheese = False
        self.toppings = 0

    def get_price(self):
        """Calculate price based on attributes"""
        price: float = 11.00
        if self.size == "large":
            price = 13.00
        if self.extra_cheese:
            price += 1.00
        price += self.toppings * 0.75
        return price


my_pizza: Pizza = Pizza()
my_pizza.toppings = 1
print(f"${my_pizza.get_price()}")

ur_pizza: Pizza = Pizza()
ur_pizza.size = "large"
print(f"Extra cheese? {ur_pizza.extra_cheese}")
print(f"${ur_pizza.get_price()}")
print(my_pizza)
```

What if we want to print a string representation of a `Pizza` object?

```
my_pizza: Pizza = Pizza()
print(f"Size: {my_pizza.size}")
print(f"Extra cheese? {my_pizza.extra_cheese}")
print(f"Toppings: {my_pizza.toppings}")
```

printing all attributes of the `my_pizza` object

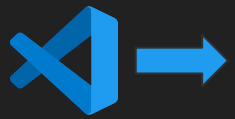
```
print(my_pizza)
```

What does this print?

```
<__main__.Pizza object at
0xffffbb399640>
```

What does the `0xffffbb399640` mean?!

Surely, there must be a better way…
Perhaps with some… ✨magic✨?



Let’s try writing some class definitions with these specified attributes and methods!
Hint: to use pi, import the math module (**`import math`**)at the top of your file and write **`math.pi`** when needed

```
circ: Circle = Circle(r=2.0)
print(circ)
print(circ.area())
```

```
>>> Circle with radius 2.0
>>> 12.566370614359172
```

```
rect: Rectangle = Rectangle(w=4.0, h=5.5)
print(rect)
print(rect.area())
```

```
>>> Rectangle with width: 4.0 and height: 5.5
>>> 22.0
```